AMENDMENTS TO THE SPECIFICATION:

Please amend the specification as follows:

On page 1, please replace the paragraph beginning with "In recent years" with the following paragraph:

In recent years, computer and network technologies have improved remarkably~~, and jobs have been widely smartened accordingly~~. As a result, there have been attempts to computerize jobs in various kinds of businesses. Many jobs must not be interrupted due to failures depending on their contents, and it has recently become a common practice to build a distributed system by connecting a plurality of computers via a network. As one running method of ~~this~~ such a distributed system, multiplexing of execution of deterministic programs ~~musing~~ using ordered multicast is known.

On page 2, please replace the paragraph beginning with "Execution of a program" with the following paragraph:

Execution of a program amounts to determining the output and the next state in correspondence with the state of a computer when an input is given to the computer. A deterministic program is defined as a program which uniquely determines the output and the next state in accordance with a given input. More specifically, the deterministic program is one that references neither arbitrary values nor random numbers. A feature of the deterministic program lies in unique execution if an initial state and input sequence are determined. In the following description, a "program" in this specification indicates such a deterministic program.

On page 3, please replace the paragraph beginning with "In a distributed system," with the following paragraph:

In a distributed system, respective computers may fail independently. If the entire system does not work due to a failure of only one computer, the availability of the distributed system is lower than that of a single computer. To avoid such a situation, processes associated with the overall system must be multiplexed. By contrast, multiplexing makes the availability of the distributed system higher than that of a single computer. For example, if a distributed system constituted by 10 computers each having an availability of 99% is not multiplexed at all, the availability of that distributed system is as low as about 90%. If this system can withstand failures of up to three computers as a result of multiplexing, the system availability becomes as high as 99.9998%.

On page 3, please replace the paragraph beginning with "Since input sequences" with the following paragraph:

Since input sequences to respective programs have the same order by ordered multicast, the states of all computers are maintained equal to each other ~~ewing~~ due to the feature of the deterministic programs, and all output sequences are equal to each other. That is, execution of programs is multiplexed.

On page 4, please replace the paragraph beginning with "As the system is" with the following paragraph:

As the system is premised on ~~that~~ all computers may each fail and may come to a halt ~~anytime~~ at any time, the overall process must not depend on a specific computer to establish multiplexed processes. Therefore, the following points must be noted.

On page 4, please replace the paragraph beginning with "For example, a simple algorithm" with the following paragraph:

For example, a simple algorithm in which input reception is fixed at a specific computer to determine the order of inputs by temporarily transferring all inputs to that computer, and the inputs are delivered in that order, cannot be used. With this algorithm, if the computer at which input reception is fixed has failed and come to a halt, the order of inputs cannot be determined at that time.

On page 5, please replace the paragraph beginning with "(2) Rendezvous of completion" with the following paragraph:

(2) ~~Rendezvous of completion of input delivery~~ The delivery of input data items to all computers is not fixed at a specific computer.

On page 5, please replace the paragraph beginning with "Each computer has an input" with the following paragraph:

Each computer has an input reception queue. As the first step, each computer delivers an input data item located at the head position of the input reception queue to all other computers as ~~an "input candidate" of~~ a next candidate to be processed by that computer. A computer with an empty input reception queue delivers ~~the "input~~

~~candidate"~~ an input data item obtained first ~~by~~ from another computer ~~as the first step~~ to all other ~~computer as its own "input candidate"~~ computers as a next candidate to be processed by that computer.

On page 7, please replace the paragraph beginning with "If a large time-out" with the following paragraph:

If a large time-out value is set, the time from when a failure has occurred until it is detected is prolonged. Then, detection of a failed computer is ~~waited~~ delayed in the ordered multicast protocol, and execution of ordered multicast temporarily stops during that time. As a result, execution of multiplexing temporarily stops.

On page 8, please replace the paragraph beginning with "Normally, such situation" with the following paragraph:

Normally, such a situation does not fatally influence the system. However, in a system that attaches an importance ~~on~~ to realtimeness, this requirement may not always be met. That is, the upper limit of the heartbeat time-out value is suppressed due to the presence of the realtimeness requirement, and an excessively large value cannot be set.

On page 8, please replace the paragraph beginning with "In order to achieve" with the following paragraph:

In order to achieve the above object, the present invention neither generates a split brain in principle nor ~~interrupt~~ interrupts a process upon occurrence of a failure by

ceasing failure detection. For this purpose, according to the present invention, if at least (n – f) computers are in operation, an input is delivered to these computers irrespective of the operations of other f computers.

On pages 9 and 10, please delete in its entirety the paragraph beginning with "More specifically, according to," that extends to page 10, line 15.

Please insert the following new paragraph on page 9:

More specifically, according to the present invention, there is provided a distributed system which makes n computers, which are connected via a network, operate synchronously, and provides multiplexing of at least (n – f) computers, each computer comprising: an input candidate collection device configured to collect input data items, each of which is selected as a next candidate to be processed by each of n computers, via the network; a first input candidate selection control device configured to determine whether not less than (n - f) input data items having identical contents are present, when the input candidate collection device has collected the not less than (n – f) input data items, and to settle one of the input data items having identical contents as next data to be processed, when the not less than (n – f) input data items having the identical contents are present; a second input candidate selection control device configured to determine whether the majority of collected input data items having identical contents are present, when the first input candidate selection control device determines that the not less than (n – f) input data items having identical contents are not present, and to cause the input candidate collection device to reexecute collection

after selecting the input data item as a self candidate and discard the all input data items of other candidates, when the majority of collected input data items are present; and a third input candidate selection control device configured to cause the input candidate collection device to reexecute collection after arbitrarily selecting input data item from the collected input data items as a self candidate and discarding all input data items of other candidates, when the second input candidate selection control device determines that the majority of the collected input data items are not present.

On page 10, please delete in its entirety the paragraph beginning with "In this distributed system."

On page 11, please replace the paragraph beginning with "Fig. 4 is a table for" with the following paragraph:

FIG. 4 is a table for explaining an outline of a principal part of ordered multicast executed by the distributed system of the embodiment;

On page 11, please replace the paragraph beginning with "Fig. 5 is the first flow chart" with the following paragraph:

FIG. 5 is ~~the~~ a first flow chart showing an operation sequence of a basic part for making one delivery of ordered multicast executed by the distributed system of the embodiment;

On page 11, please replace the paragraph beginning with "Fig. 6 is the second flow chart" with the following paragraph:

FIG. 6 is ~~the~~ a second flow chart showing the operation sequence of the basic part for making one delivery of ordered multicast executed by the distributed system of the embodiment;

On page 11, please replace the paragraph beginning with "Fig. 7 is the first flow chart" with the following paragraph:

FIG. 7 is ~~the~~ a first flow chart showing ~~the~~ an operation sequence for eliminating delay of multiplexing execution executed by the distributed system of the embodiment;

On page 12, please replace the paragraph beginning with "FIG. 8 is the second flow chart " with the following paragraph:

FIG. 8 is ~~the~~ a second flow chart showing the operation sequence for eliminating delay of multiplexing execution executed by the distributed system of the embodiment;

On page 12, please replace the paragraph beginning with "FIG. 9 is the third flow chart" with the following paragraph:

FIG. 9 is ~~the~~ a third flow chart showing the operation sequence for eliminating delay of multiplexing execution executed by the distributed system of the embodiment;

On page 12, please replace the paragraph beginning with "Fig. 10 is the fourth flow chart" with the following paragraph:

FIG. 10 is ~~the~~ a fourth flow chart showing the operation sequence for eliminating delay of multiplexing execution executed by the distributed system of the embodiment; and

On page 12, please replace the paragraph beginning with "A embodiment of the present invention" with the following paragraph:

~~A~~ An embodiment of the present invention will be described hereinafter with reference to the accompanying drawings.

On page 14, please replace the paragraph beginning with "As shown in FIG. 1," with the following paragraph:

As shown in FIG. 1, a distributed system 1000 is multiplexed by n computers 100, each of which is connected to a plurality of clients 2000 via an external network A. These computers 100 are connected via an internal network B. Each computer 100 in this distributed system 1000 processes input packets (inputs 1) received from the clients 2000 via the external network A or input packets (inputs 2) received from other computers 100 via the internal network B in the same order as in other computers 100. An input packet (input 1) from each client 2000 is input to one of n computers 100. An example of input packets (inputs 2) from other computers 100 is an input packet which is generated by another arbitrary computer 100, i.e., in a local environment in the distributed system 1000, and ~~process~~ processing of which is requested via the internal network B.

On page 14, please replace the paragraph beginning with "Output packets generated by this process" with the following paragraph:

Output packets generated by this process are returned to the clients ~~200~~ 2000 via the external network A (outputs 1), or are returned to other computers 100 via the internal network B (outputs 2).

On page 15, please replace the paragraph beginning with "The respective building" with the following paragraph:

The ~~respective building~~ components of the ordered multicast unit 2 will be explained below.

On page 18, please replace the paragraph beginning with "An outline of principal" with the following paragraph:

An outline of a principal part of ordered multicast executed by the ordered multicast unit 2 will be explained below with reference to FIG. 4.

On page 18, please replace the paragraph beginning with "Assume that computers (1) and (2)," with the following paragraph:

Assume that computers (1) and (2), computer (3), and computer (4) respectively select A, B, and C as input candidates in ~~the~~ a first step. Assume that computer (1) collects input candidate A of computer (2) and input candidate B of computer (3) in ~~the~~ a second step. That is, computer (1) collects (n – f) candidates including the self candidate and other candidates. At this time, computer (1) attempts to check input

candidates before collecting the input candidate of computer (4). However, since the input candidates do not include (n – f) identical candidates, computer (1) executes re-selection of an input candidate. Upon re-selection, if given candidates account for a majority of the collected input candidates, that candidate is selected; if such candidate is not present, one of the collected candidates is randomly selected. In this case, since A accounts for a majority, computer (1) re-selects A as a self candidate in ~~the~~ a third step.

On page 19, please replace the paragraph beginning with "Assume that computer (1) collects input candidate A" with the following paragraph:

Assume that computer (1) collects input candidate A of computer (2) and input candidate A of computer (4) in ~~the~~ a fourth step. That is, computer (1) has collected (n – f) candidates including the self candidate and other candidates again. At this time, computer (1) attempts to check input candidates before collecting the input candidate of computer (3). In this case, since there are (n – f) candidates A, computer (1) determines A as an input in ~~the~~ a fifth step.

On page 20, please replace the paragraph beginning with "Assume that computer (2) selects input candidate A" with the following paragraph:

Assume that computer (2) selects input candidate A of computer (1) and input candidate A of computer (3) in ~~the~~ a sixth step. In this case, since input candidate A of computer (1) is already not a candidate but a settled input, computer (2) determines A as an input in ~~the~~ a seventh step.

On page 25, please replace the paragraph beginning with "(Algorithm 1) to (algorithm 4) form basic part" with the following paragraph:

(Algorithm 1) to (algorithm 4) form a basic part for making one delivery of ordered multicast. The conventional system repeats the process until all normal computers match, but this distributed system repeats the process until (n – f) computers match.

On page 28, please replace the paragraph beginning with "This delay is generated when" with the following paragraph:

This delay is generated when multiplexing is executed by ~~computers~~ more than (n – f) computers. The delayed computer is not necessary for multiplexing at that time. However, such computer is required to continue multiplexing when the leading computer has failed and halted. That is, in such case, the delayed computer must catch up with the final input order number.

On page 30, please replace the paragraph beginning with "FIGS. 5 and 6 are flow charts" with the following paragraph:

FIGS. 5 and 6 are flow charts showing the operation sequence of a basic part for making one delivery of ordered multicast.

On page 32, please replace the paragraph beginning with "On the other hand" with the following paragraph:

On the other hand, if (n – f) or more identical candidates are not present (NO in step A2 in FIG. 5), the ordered multicast unit 2 checks if identical candidates that hold

the majority are present (step A5 in FIG. 5). If such candidates are present (YES in step A5 in FIG. 5), the ordered multicast unit 2 selects that candidate as a self candidate, and sends that self candidate to all other computers (step A6 in FIG. 5). Then, the ordered multicast unit 2 repeats the process from step A1. At this time, the ordered multicast unit 2 discards all stored other candidates. On the other hand, if identical candidates which hold the majority are not present (NO in step A5 in FIG. 5), the ordered multicast unit 2 randomly selects a self candidate, and sends that self candidate to all other computers (step A7 in FIG. 5). Then, the ordered multicast unit 2 repeats the process from step A1. At this time as well, the ordered multicast unit 2 discards all stored other candidates.

On page 35, please replace the paragraph beginning with "As shown in FIG. 11" with the following paragraph:

As shown in FIG. 11, this computer 100 comprises a virtual time counter 7 for counting the time used upon executing the application program 3, and a virtual time manager 8 for managing this virtual time counter 7 in addition to a system ~~block~~ clock 6 that counts a so-called "system time".

On page 36, please replace the paragraph beginning with "Upon starting up" with the following paragraph:

Upon starting up the computer 100, ~~The~~ the initial condition of the virtual time counter 7 is made to be an un-setup condition.

On page 36, please replace the paragraph beginning with "After that, the virtual time manager" with the following paragraph:

After that, the virtual time manager 8 casts an input packet with high priority for giving an increment timing of the virtual time counter 7 to the input reception queue 1, e.g., every second. This input packet is immediately fetched by the ordered multicast unit 2 due to its priority level, and is set as the self candidate, which is an input candidate of the self computer. As a result, the input settlement checking unit 26 settles this input packet as the next input, casts it to the virtual time manager 8. At this time, in other computers 100, the input settlement checking unit 26 settles this input packet as the next input, and casts it to the virtual time manager 8.

On page 36, please replace the paragraph beginning with "Upon receiving this input packet" with the following paragraph:

Upon receiving this input packet, the virtual time manager 8 increments the virtual time counter 7 by a predetermined value (normally, 1 sec). ~~the~~ The virtual time manager 8 also removes this input packet from the reception queue, and prepares for ~~re-cast~~ re-casting this input packet to the input reception queue 1 in one second from this moment. At this time, in other computers 100, if this packet is present in the input reception queue 1, the virtual time manager 8 removes this input packet from the reception queue, and prepares for ~~re-cast~~ re-casting this input packet to the input reception queue 1 in one second from this moment. That is, ~~Even~~ even if this input packet casted by other virtual time manager 8 is settled, the virtual time manager 8 processes this input packet just ~~like~~ as though casted by itself, and ~~do~~ does not cast this

input packet until one second later from this moment. In this way, all the computers 100 can count the virtual time at the same timing.

On page 37, please replace the paragraph beginning with "On the other hand" with the following paragraph:

On the other hand, the virtual time manager 8 casts an input packet with high priority for giving a comparison timing between the system time and virtual time to the input reception queue 1 when starting up the computer 100 and , e.g., every hour. At this time, the virtual time manager 8 stores the system time counted by the system clock 6 in this input packet. Since this input packet is sent to other computers by the protocol data exchange unit 23, other computers consequently receive ~~receives~~ the system time message of the self computer. Likewise, when the virtual time manager 8 of another computer generates this input packet, the self computer receives the system time message of that computer.

On page 37, please replace the paragraph beginning with "Upon receiving the input packet" with the following paragraph:

Upon receiving the input packet, the virtual time manager 8 compares the system time of the self computer or another computer stored in that input packet with the virtual time counted by the virtual time counter 7. If the virtual time counter 7 is in an un-setup condition, the virtual time manager 8 stores the system time counted by the system clock 6 in the virtual time counter. On the other hand, if the virtual time counter 7 is not in an un-setup condition and the system time leads the virtual time, the virtual time

manager 8 executes a process for advancing the virtual time faster than a normal state. For example, the increment width of the virtual time counter 7 upon casting an input packet that gives the increment timing of the virtual time counter is set to be larger than the normal state.